Algorithm Theoretical Basis Document for a PLI field for the Nordic Sea

STARS deliverable document D-8, ATBD-1

Version 1.0

January 2011

Magnus Drivdal and Øyvind Saetra
Norwegian Meteorological Institute

Documentation Change Record

Document version	Date	Description
V 1.0	19-01-2011	First version sent to ESA

CONTENTS

1. Introduction

In the theoretical investigation of *Emmanuel (1986),* the polar low is assumed to work as a Carnot engine. Under this assumption, the energy to the polar low comes from the air being heated in the boundary layer and cooled at low temperatures at the cloud top level in the atmosphere *(Linders et. al. 2010).* The interpretation of polar lows being dependent on temperature differences is one motivation for the development of a Polar Low Indicator (PLI) field.

The general idea of the PLI field is that a large difference in the temperature in the 500hPA surface and the SST, as the Carnot engine theory suggests, provides favorable conditions for polar lows to develop and intensify. Identifying areas with such conditions would then be helpful for the forecast of polar lows.

The formation of polar lows always take place during marine cold-air outbreaks. This happens when cold and very stable air over the Arctic ice mases are set into motion by the large scale atmospheric flow in such a way that i become exposed to the relatively warm open ocean.

The strength of the geostrophic wind is proportional to the pressure gradient, and hence the surface pressure field can be used to determine a marine cold-air outbreak. The goal of this algorithm is to store the fields of SST minus the temperatures in the 500hpa-surface (SST-T500) and the surface pressure in one NetCDF file for each day in the STARS-DAT period. Since the polar lows have been cataloged in STARS-DAT for this period, it is possible evaluate to which extent these files can be used in predicting polar lows.

The idea of the PLI is that the Atmospheric Heat Potential (AHP) must be sufficiently large for polar lows to be maintained. Here the AHP is defined as the difference in temperature between the sea surface and the 500hPa surface. Suggested threshold values for polar lows to exist are around 42 degrees. However, as mentioned above. An additional necessary conditions for polar low development is the presence of a marine cold-air outbreak. This trigger deep-convection and the formation of large cumulonimbus clouds. When the AHP in this situation is sufficiently large, polar low development may be triggered.

The intention of this document is to outline the concept of a Polar Low Indicator (PLI) based on information on sea-surface and atmospheric temperatures together with information on sea-surface pressure to determine whether or not a cold-air outbreak is on.

1.1 Reference Documents

[RD-1] Eastwood, S. and M. Drivdal (2011): STARS-DAT User Manual v2.0. STARS deliverable document D-10.

[RD-2] Drivedal, M., G. Noer and Ø. Saetra (2011): Polar Low Indicator (PLI) development, implementation and validation report. STARS deliverable document D-9.

2. Algorithm overview

2.1 State of the art for storm intensification fields

The tropical cyclone heat potential (or sometimes referred to as hurricane heat potential) is defined as the integrated vertical temperature from the sea surface to the depth of the 26°C isotherm (Leiper and volgenau, 1973; Law and Hobgood, 2007). The is based on the understanding that hurricane intensity is limited by the upwelling of cold water normally observed over waters with a shallow thermocline. As mentioned previously, deep cores of warm waters act as insulation to entrainment of the cold waters under the thermocline and may lead to extremely intense hurricanes. For this purpose, the heat potential as defined above is particularly attractive for hurricane forecasting is the fact that relatively reliable estimates can be made from the sea-surface height anomaly observed by satellite altimeter sensors (Lin et al. 2008). Another point to mention here is that hurricanes are exclusively driven by energy supply from the ocean surface. Virtually no baroclinicity is found in the tropical atmosphere where hurricanes occur. The integrated heat contained in the upper ocean therefore makes a very unique indicator of the possible intensity. This is probably very much in contrast to polar lows, for which the debate about relative importance of baroclinicity versus convective release of latent heat has been going on for as long as polar lows have been studied. The view that polar lows are mixed systems of baroclinic and diabatic processes implies that the ocean temperature is not to the same degree controlling the cyclone intensity as in the case for hurricanes.

Bracegridle and Gray (2008) found that the difference between the wet-bulb potential temperature at 700 hPa and the sea surface temperature (SST) is an effective discriminator between the atmospheric conditions associated with polar lows and other cyclones in the Nordic seas. We will refer to this as the Atmospheric Heat Potential (AHP). A verification study shows that the objective identification method is reliable in the Nordic seas region. Kolstad et al. (2009) investigated the spatial and temporal distributions of marine cold air outbreaks (MAOC) over the northern North Atlantic using re-analysis data for the period from 1958 to 2007. They used a simple index for identifying cold air outbreaks: the vertical potential temperature gradient between the sea surface and 700 hPa. It was found that atmospheric temperature variability is considerably more important than the sea surface temperature variability in governing both the seasonal and the inter-annual variability of cold air outbreaks. Furthermore, a composite analysis revealed that a few well-defined and robust synoptic patterns are evident during cold air outbreaks in winter. Over the Labrador and Irminger Seas the MCAO index was found to have a correlation of 0.70 with the North Atlantic Oscillation index, while over the Barents Sea a negative correlation of 0.42 was found.

2.2 Instrument and data characteristics

For the temperature at the 500hPa surface and the surface pressure fields, the 00:00UTC HIRLAM analysis will been used. This is the operational forecast model at met.no. More details on this model can be found in the STARS-DAT User Manual

[RD-1], and on http://www.hirlam.org. These data are available in the NetCDF format in STARS-DAT.

The SST data are collected from the GHRSST LTSRF archive at http://ghrsst.nodc.noaa.gov, and have been re-gridded to the STARS-DAT grid. A more in-depth description of this can also be found in the STARS-DAT User Manual [RD-1].

2.3 PLI retrieval strategies

The polar low indicator (PLI) is calculated on a daily basis for a set of preselected sectors (described in more details under Section 3). In each of these sectors three data fields are needed: surface analysis of mean sea level pressure (MSLP), temperature of the 500hPa (T500) surface and daily average of the sea-surface temperature (SST). In a forecasting situation, the input data must be based on model forecasts of MSLP pr and T500 temperatures

To retrieve the temperature -and pressure data, which are both available in the NetCDF format, the statistical tool R is used (http://cran.r-project.org/). This enables the efficient collection and handling of the gridded data as matrices. Simple operations on these matrices is also easy to implement in R, so that the SST-T500 matrix can be computed directly when generating the new PLI-files.

3. Algorithm Description

3.1 Theoretical description

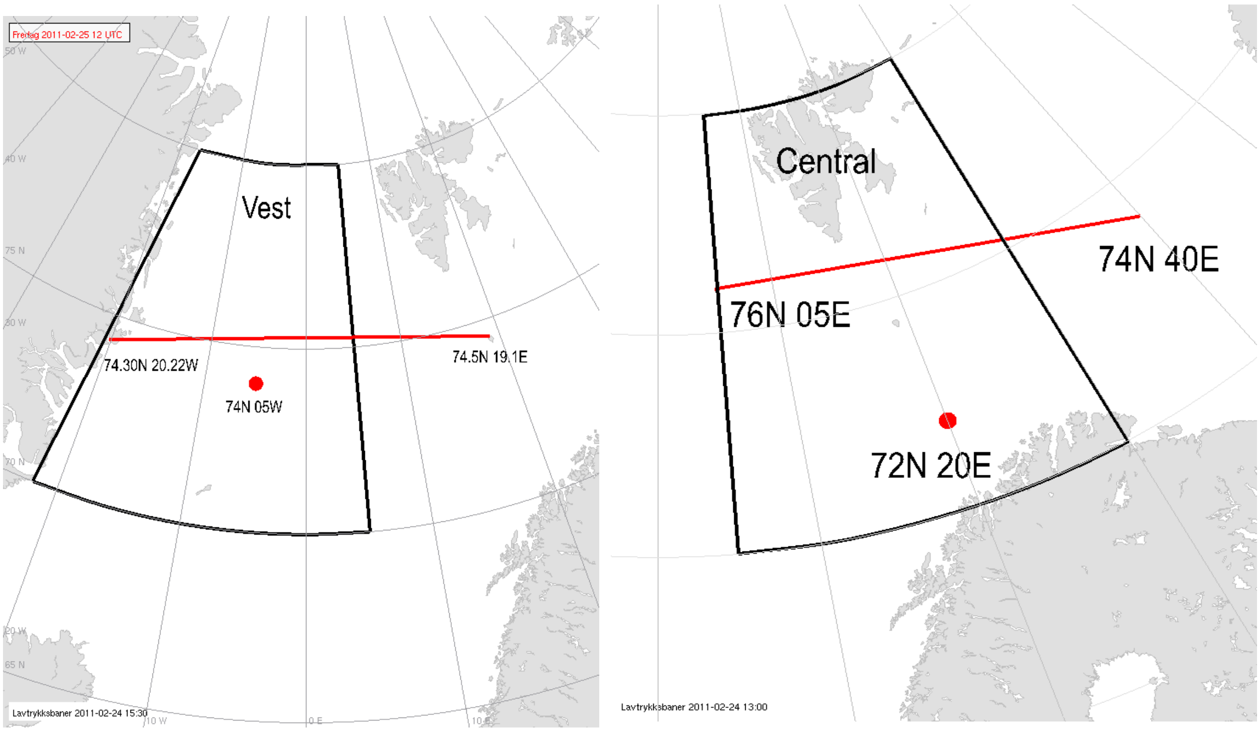
For the PLI to be meaningful a cold-air outbreak must coexists with the AHP above a AHP value above the threshold value. The threshold value is determined by maximizing the Probability Of Detection (POD) in [RD-2]. To co-locate large values of AHP with marine cold-air outbreaks, the Nordic Seas are divided into sectors where polar lows are known to form when the cold-air outbreaks is from a specific direction. The sectors used here is shown in Figure 1. In Figure 2, a flow diagram on how the PLI combine input data and determined whether or to a polar low is to be foretasted.

For each day, the direction of the geostrophic wind in the atmospheric boundary layer is estimated by calculating the pressure difference along the red line in each sector. If no cold-air outbreak is detected, the PLI issues a 'no polar low' forecast. If a marine cold-air outbreak is detected, the AHP is calculated as the temperature difference between the sea-surface and the 500hPa surface. If the AHP exceeds the determined threshold value, calculated in a point centrally located inside the sector, 'possible polar low' is foretasted.

In the HIRLAM NetCDF files, only the air potential temperatures are given, and hence a transformation into temperatures is necessary. The mathematical details of this is presented below. Using only the 00:00 UTC HIRLAM analysis then yields a matrix with the temperatures at 500hPa (T500) in the grid.

The AMSRE SST data are available from several orbits each day, and often the measurements overlap each other in the grid. For this reason, the mean SST is calculated in the grid-points with more than one measurement. The final SST-matrix then represents a daily mean. Basic matrix operations to calculate the SST-T500 temperature differences are easily implemented in R (see the script in Appendix A).

For each day in the STARS-DAT period, a PLI-file in the NetCDF format is then produced. This contains the PLI field (SST-T500), which is also called the Atmospheric Heat Potential, and the surface pressure field. The pressure field is copied directly from the 00:00 UTC HIRLAM analysis. These PLI files will then be sufficient to implement the further PLI analysis.



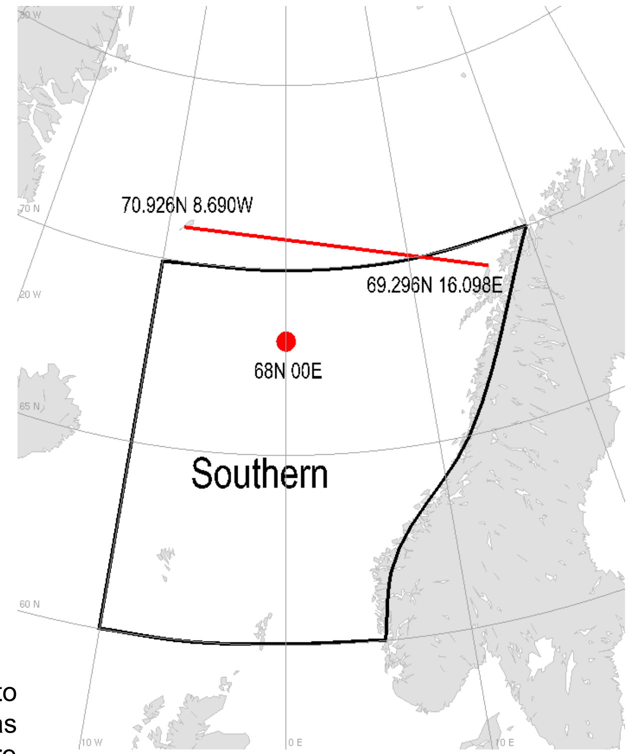


Figure 1: The four sectors that were defined to identify cold-air outbreaks. In each sector data was collected from the PLI-files for each day in the entire STARS-DAT period. The red dots indicate where the temperature differences between the SST and the 500hPa-surface collected. The pressure data were collected from the ends of each red line, and the pressure difference was calculated by subtracting the surface pressure at the easternmost ends from the surface pressure at the westernmost ends (i.e. positive difference corresponds to southerly geostrophic wind).

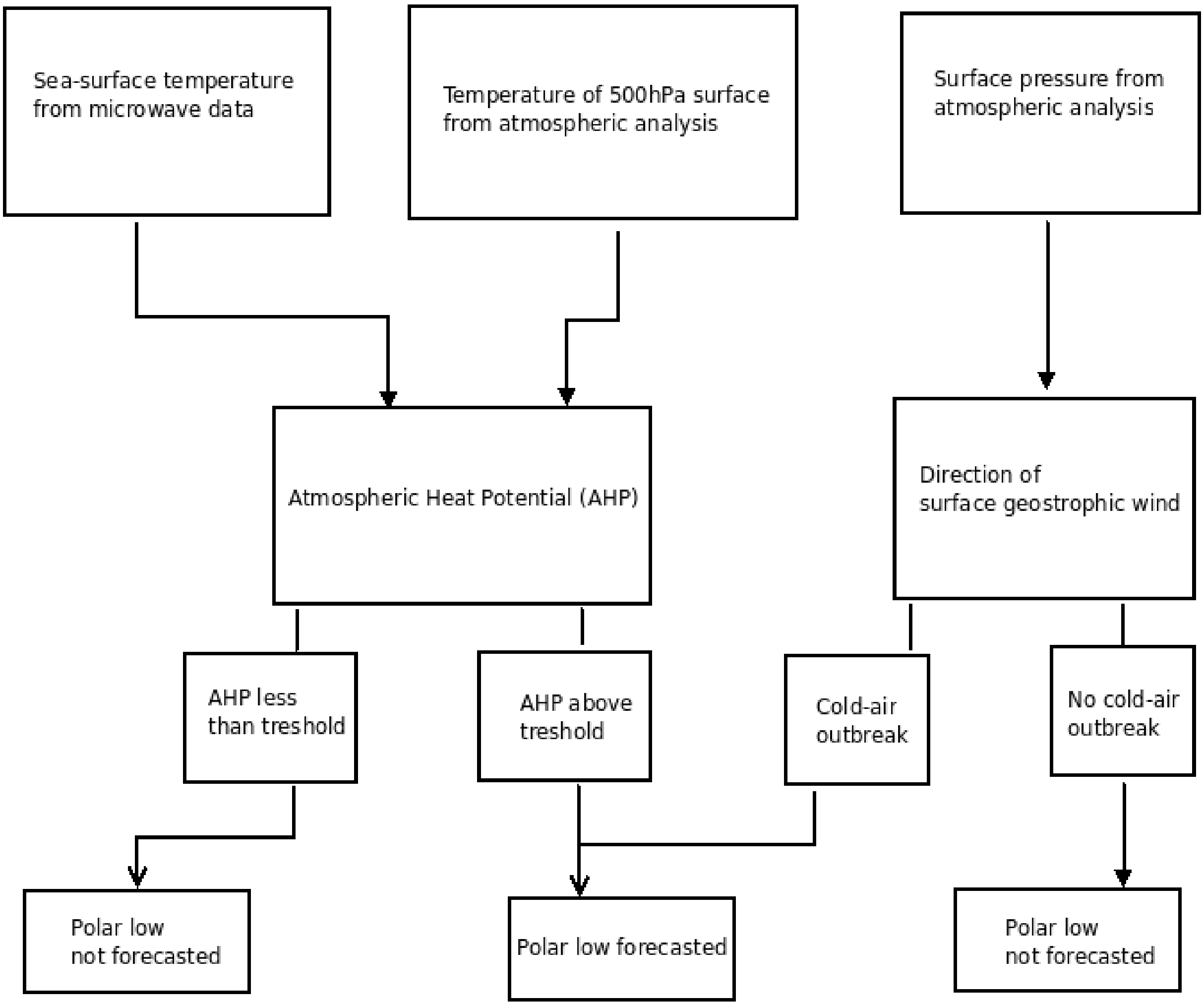


Figure 2: Flow diagram showing how the atmospheric and sea-surface data are combined to determine whether or not the PLI shall forecast polar lows.

3.2 Mathematical description

The potential temperature is the temperature a parcel of air at pressure P would acquire if brought adiabatically to a standard reference pressure P_0. This is described by the formula below

formula $\theta = T\left(\frac{P_0}{P}\right)^{\frac{R}{c_p}}$,

where θ is the potential temperature, T is the temperature, R is the gas constant of air and c_p is the specific heat capacity at constant pressure. Since we are interested in the temperature rather than the potential temperature, we need the inverse formula:

formula $T = \theta\left(\frac{P}{P_0}\right)^{\frac{R}{c_p}}$

In HIRLAM, the fractionformula $\frac{R}{c_p}$ is set to 0.286, and the standard reference pressure P_0 is 1000hPa.

The geostrophic wind component normal to the preselected lines in the relevant sector is calculated as:

$$U_g = \frac{1}{\rho f} \frac{P_2 - P_1}{L} \quad ,$$

Where U_g is the geostrphic wind component normal to the line, ρ is the air density, f is the Coriolis parameter P_2 is the surface pressure at the westernmost point and P_1 is the pressure at the easternmost position. L is the distance between the two pressure points. At the Northern Hemisphere a negative value of the geostrophic wind indicates southernly winds. A geostrophic magnitude of more than 10m/s combined with the appropriate wind direction determines the cold-air outbreak. The Atmospheric Heat Potential (APH) is then calculated as

$$AHP = T500 - SST$$

in a preselected point inside the relevant sectors. The threshold value of the AHP is expected to be in the range 42-46 degrees Celsius.

3.3 Assumptions and limitations

The analysis presented above is based on the assumption that the microwave satellite data give correct information of the sea-surface temperatures. Furthermore, it is assumed that the HIRLAM analysis give accurate values for the 500hPa temperature and the mean sea-level pressure. The Hirlam assimilation system at met.no is blending with the 4D-VAR analysis from ECMWF. Here high-spectral resolution satellite data are assimilated, yielding tropospheric temperatures in areas with spares network of radiosondes for atmospheric profiling. The large scale wind pattern in the planetary boundary layer is calculated from atmospheric analysis where synoptic surface observations have been assimilated. The atmospheric analyses are continuously validated (see for instance http://www.ecmwf.int/products/forecasts/d/charts/medium/verification/) and generally considered to be of good quality. Under most conditions the temperature differences and geostrophic wind directions based on these data should be fairly accurate.

The polar low indicator implicitly assumes that the 500hPa surface is representative for the vertical level where cooling by long-wave radiation takes place. Normally, this happens at the cloud top level. Typically, deep convective cloud towers, which are more or less a requirement for a system to be defined as a polar low, penetrates the whole troposphere. Accordingly, cloud-top levels are usually found slightly below the tropopause. During cold-air outbreaks the tropopause is usually at about 500hPa, which somehow lower than the tropopause level found in synoptic low-pressure systems. An example of this is shown in Figure 3, which depicts the relative humidity through a section of a polar low. The polar low was observed by dropsondes on 4 March 2008 during the IPY-THORPEX field campaing (Linders and Seatra, 2010). The red colors indicate high values of relative humidity. The deep convective clouds surrounding the clear eye is clearly seen. Note that cloud tops coincided very well with the 500hPa level.

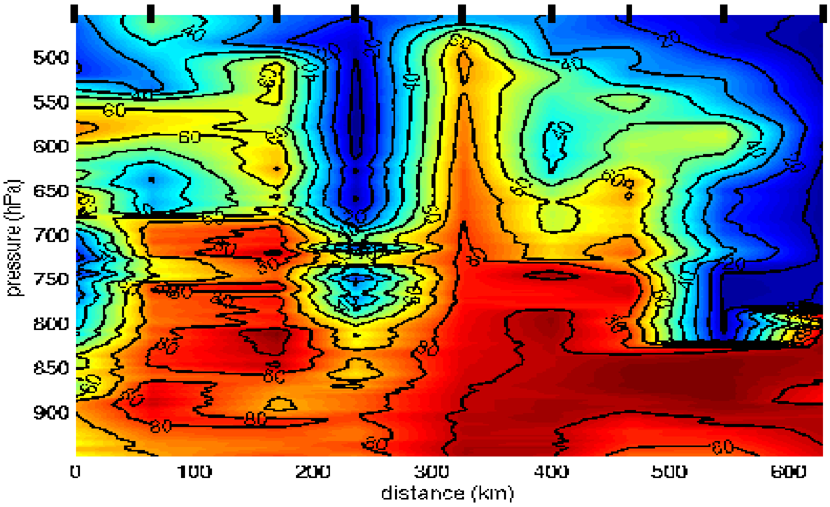


Figure 3: Relative humidity from dropsondes taken through a section of a polar low on 4 March 2008 (Linders and Saetra, 2010)

The existence of a marine cold-air outbreak is determined from the pressure gradient (geostrphic wind) along a number of predefined lines. The selections cover most of the geostrophic wind directions when cold-air outbreaks occur in these areas. There may be rare exceptions from this particular selection. Still, we believe the selection used here are representative for most cold-at outbreak situations.

The PLI is based on the assumption that polar lows occur when the AHP is large in combination with a cold-air outbreak. This is by no means a guarantee the polar lows will develop. Polar low formation is a complex process involving highly non-linear dynamics. It is also well known that the atmosphere has a degree of chaotic behavior where cyclone formations may depend on small atmospheric perturbations. All the necessary atmospheric and oceanic conditions may be present and still, no polar low observations are made. This is perhaps the major limitations of a detection system based on an environmental indicator and will necessarily lead to over-forecasting when used without additional judgment by forecasters. Accordingly, a high number of false alarm is expected when validated in an 'automatic' configuration.

4. Demonstration results

4.1 Results of the approach

Through the approach described above, PLI fields can be plotted directly in R. The results for the first day of the polar lows in the STARS-DAT data set show that there is a clear tendency for polar lows to occur in areas with temperature differences greater than 42-46 degrees (Celsius/Kelvin). These images are available on the on-line PHP interface described in the STARS-DAT user manual [RD-1] (http://projects.met.no/stars/view_stars-dat.php). An example of this is given in figure 4. The greatest challenge for the application of the PLI is thereby to investigate if temperature differences of these magnitudes often occur without the development of polar lows. Using the surface pressure field, the strength -and direction of the geostrophic wind can be used as an additional requirement to identify cold-air outbreaks. Hopefully this will lower the amount of false alarms. Suitable threshold values will be found by both using the experience of the forecasters at the meteorological institute in Tromsø, and by a simple process of trial -and error.

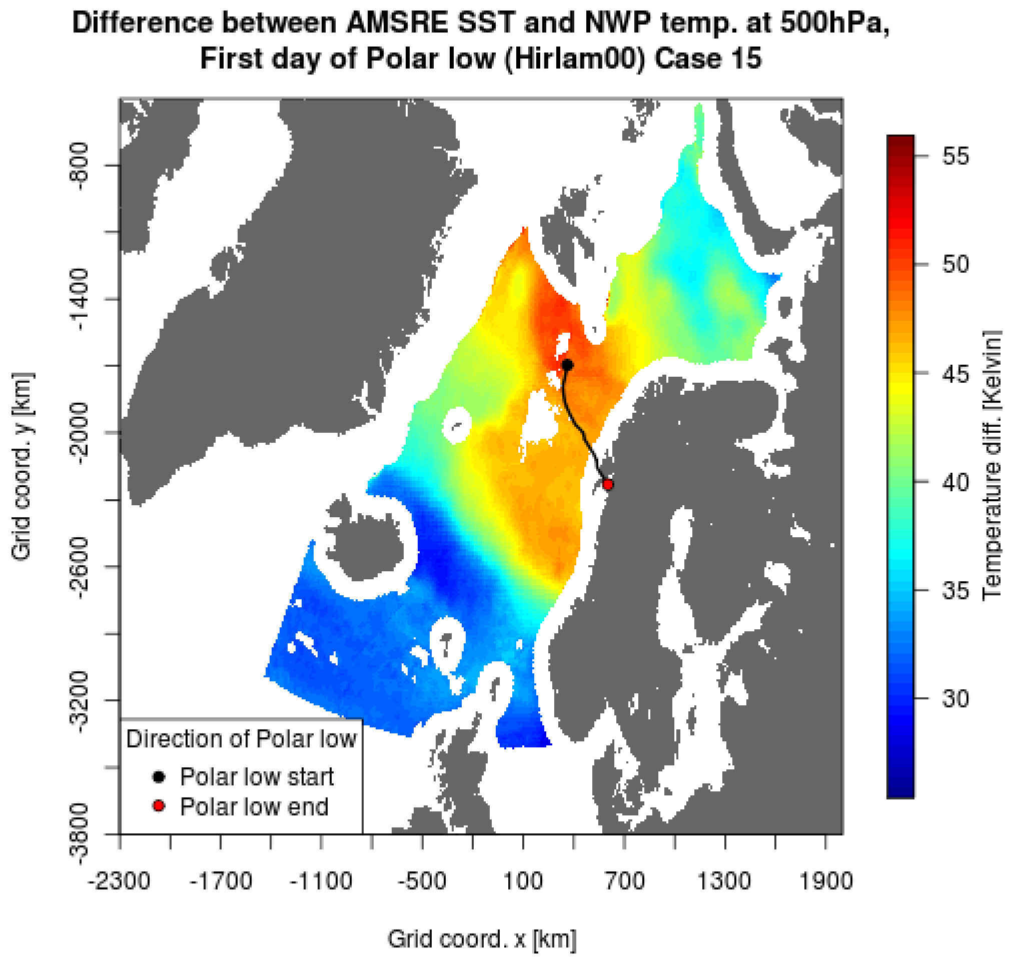


Figure 4: The AHP for 6 April 2007. The AHP is shown as color shadings with red indicating high values. A track of the polar low that hit the coast of Northern Norway is shown as a black solid line.

The example given in Figure 4 depicts the polar low track of an event that hit the coast of Norway on 6 April 2007 on top of a color shaded field of the AHP. Note that the polar low in this case coincide closely with areas of large AHP values.

4.2 Validation plan

Running through the whole STARS-DAT period and cataloging all events with temperature -and pressure differences exceeding the threshold values will make it possible to quantify how often these events coincide with the polar low events. The number of situations where only one of the two events occur will reveal the potential of the temperature difference as an indicator of polar lows.
Since this procedure will result simple "yes" or "no" forecasts and "yes" or "no" observations, the simple two by two contingency table and associated statistics can easily be found. A detailed description of how this is done can be found in Wilks (1995), ch. 7.2.

4.3 Further work

If the investigations prove that the temperature difference between the SST and the 500hPa surface is a good indicator of polar lows, this can be used to develop operational tools for forecasting.

5. Conclusions and recommendations

The data required for the PLI are available in STARS-DAT in the NetCDF format, and can easily be accessed and handled by R (http://cran.r-project.org/). Since only basic mathematical methods are necessary to make the PLI field, the implementation in R should be straightforward.

One of the main challenges will be the validation of the PLI as a tool for forecasting polar lows. With all observed polar lows over an extended period available in the STARS-DAT data set, the validation process should give a good indication of the potential of the PLI.

6. References

Bracegridle, T. and S. Gray. 2008. An objective climatology of the dynamical forcing of polar lows in the Nordic Seas. *Int. J. Climatol*. **28**. 1903-1919.

Emanuel K. A. 1986: A two stage air-sea interaction theory for polar lows. In: Proceedings of the international conference on polar lows, Oslo, Norway, 20-23 May 1986. The Norwegian Meteorological Institute, pp. 187-200.

Law, K. T., and J. S. Hobgood. 2007. A Statistical Model to Forecast Short-Term Atlantic Hurricane Intensity. *J. Atmos. Sci.* **22***.* 967-980.

Leiper D., and D. Volgenau, 1972: Hurricane heat potential of the Gulf of Mexico. *J. Phys. Oceanogr.*, **2**, 218–224.

Lin, I.-I, C,-C Wu, I.-F. Pun and D.-S Ko. 2008. Upper-Ocean Thermal Structure and the Western North Pacific Category 5 Typhoons. Part I: Ocean Features and the Category 5 Typhoons' Intensification. *Mon. Wea. Rev*. **136**, 3288-3306.

Linders, T., T. Bracegridle and Ø. Saetra, 2010: Limited polar low sensitivity to SST. In: Quarterly Journal of the Royal Meteorological Society, accepted September 2010.

Linders, T. and Ø. Saetra, 2010: Can CAPE Maintain Polar Lows?, *J. Atmos. Sci.* **67***,* 2559-2571.

Kolstad, E. W, 2005: A new climatology of favourable conditions for reverse-shear polar lows. Bjerknes Centre for Climate Research, Allegaten 70, 5007 Bergen, Norway.

Wilks Daniel S. 1995: Statistical Methods in the Atmospheric Sciences. Vol. 59 in International geophysics series, Academic Press.

7. Appendix A – R script for PLI calculation

```
library(ncdf)

startstring <- "/fou/stars/data/STARS-DAT/gridded"

#Landmask:
landm.nc = open.ncdf("/fou/stars/data/STARS-DAT/static/lmask_eurarc_stere_20_stars.nc")
xc_lm = get.var.ncdf(landm.nc,"xc")
yc_lm = get.var.ncdf(landm.nc,"yc")
landmask = get.var.ncdf(landm.nc,"land_area_fraction")
landmask[landmask <= 0] = NA
close.ncdf(landm.nc)

#The                                                                                    autumn
06!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
start.time.06 <- ISOdatetime(2006,10,01,0,0,0,tz="UTC")
startjulday.06 <- strptime(start.time.06,"%Y-%m-%d")$yday+1
endjulday.06 <- strptime(ISOdatetime(2006,12,31,0,0,0,tz="UTC"),"%Y-%m-%d")$yday+1

for( i in startjulday.06:endjulday.06) {
  cat(paste("processing julian day ",i," in 2006\n",sep=""))

  epocl <- start.time.06 + 24*60*60*(i-startjulday.06)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")

  #SST-------------------------------------------------------------------------------------------------------
----------------------------------
  filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
   if(length(filelist_sst_1) != 0) {
    for(j in 1:length(filelist_sst_1)) {

      temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

      xc <- get.var.ncdf(temp_sst_1.nc,"xc")
      yc <- get.var.ncdf(temp_sst_1.nc,"yc")
      sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                                  #get fillvalue for SST
              Fillval_sst      =  att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

    #set sst with fillvalue to NA's
      sst[sst==Fillval_sst] <- NA

                                  #Total vector with SST:
      if( j == 1) {
        SST_tot <- as.vector(sst)
      }
      else{
```

```
      SST_tot <- rbind(SST_tot,as.vector(sst))
     }
     close.ncdf(temp_sst_1.nc)
    }

                                #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc
    if(length(filelist_sst_1) != 1) {
     SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
    }
    else{
     SST_mean <- matrix(SST_tot,dim(sst))
    }

  }
  else {
   SST_mean <- matrix(nrow=2150,ncol=1650)
   cat(paste("No SST-files for julian day ",i," 2006","\n",sep=""))
  }

                                              #T500                    /                    T700
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------

  file.hir.1 <- paste(file.loc1,"/","h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

  if( file.exists(file.hir.1) ){
    temp_hir_1.nc <- open.ncdf(file.hir.1)
    timevector <- get.var.ncdf(temp_hir_1.nc,"time")
    pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")

   #Temp_diff
    Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value

                                                                  pot.temp500                <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp500[pot.temp500 == Fillval_temp] <- NA
    temp500 <- pot.temp500*(500/1000)^0.286
    difftemp500 <- SST_mean - temp500

                                                                  pot.temp700                <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp700[pot.temp700 == Fillval_temp] <- NA
    temp700 <- pot.temp700*(700/1000)^0.286
    difftemp700 <- SST_mean - temp700

    #Pressure (Maybe only in points?)
                                                              Sea.level.pressure              <-
get.var.ncdf(temp_hir_1.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),length
(yc_lm),1))

    close.ncdf(temp_hir_1.nc)
   }
  else{
```

```
    difftemp500 <- matrix(nrow=2150,ncol=1650)
    difftemp700 <- matrix(nrow=2150,ncol=1650)
    Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
  }

  missval = 1.0e10
  difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
  difftemp500[is.na(difftemp500)] = missval
  difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
  difftemp700[is.na(difftemp700)] = missval

    # Save as netcdf file
  outfile = paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")
  cat(paste(outfile,"\n"))

  xc <- dim.def.ncdf("xc","km",1:2150)
  yc <- dim.def.ncdf("yc","km",1:1650)
  varAHP500 <- var.def.ncdf("ahp_500", "degrees", list(xc,yc), missval, longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
  varAHP700 <- var.def.ncdf("ahp_700", "degrees", list(xc,yc), missval, longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
  varPres <- var.def.ncdf("sea_level_pressure", "Pa", list(xc,yc),-1, longname="Sea level pressure", prec="float")

    # Create a netCDF file with this variable
  ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )

     # Write some values to this variable on disk.
  put.var.ncdf( ncnew, varAHP500, difftemp500)
  put.var.ncdf( ncnew, varAHP700, difftemp700)
  put.var.ncdf( ncnew, varPres, Sea.level.pressure)

  close.ncdf(ncnew)


}


#The Spring 07!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
start.time.07 <- ISOdatetime(2007,1,1,0,0,0,tz="UTC")
startjulday.07 <- strptime(start.time.07,"%Y-%m-%d")$yday+1
endjulday.07 <- strptime(ISOdatetime(2007,04,30,0,0,0,tz="UTC"),"%Y-%m-%d")$yday+1

for( i in startjulday.07:endjulday.07) {
  cat(paste("processing julian day ",i," in 2007\n",sep=""))

  epocl <- start.time.07 + 24*60*60*(i-startjulday.07)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")
```

```
 #SST------------------------------------------------------------------------------------------------------------
------------------------------------
 filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
  if(length(filelist_sst_1) != 0) {
   for(j in 1:length(filelist_sst_1)) {

     temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

     xc <- get.var.ncdf(temp_sst_1.nc,"xc")
     yc <- get.var.ncdf(temp_sst_1.nc,"yc")
     sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                                   #get fillvalue for SST
          Fillval_sst      =  att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

   #set sst with fillvalue to NA's
     sst[sst==Fillval_sst] <- NA

                                   #Total vector with SST:
     if( j == 1) {
       SST_tot <- as.vector(sst)
     }
     else{
       SST_tot <- rbind(SST_tot,as.vector(sst))
     }
     close.ncdf(temp_sst_1.nc)
   }

    if(length(filelist_sst_1) != 1) {
     SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
   }
   else{
     SST_mean <- matrix(SST_tot,dim(sst))
   }
                                   #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc

 }
 else {
   SST_mean <- matrix(nrow=2150,ncol=1650)
   cat(paste("No SST-files for julian day ",i," 2007","\n",sep=""))
 }

                                   #T500                    /                    T700
-----------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------

 file.hir.1 <- paste(file.loc1,"/","h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

 if( file.exists(file.hir.1) ){
    temp_hir_1.nc <- open.ncdf(file.hir.1)
    timevector <- get.var.ncdf(temp_hir_1.nc,"time")
    pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")
```

```
  #Temp_diff
   Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value

   pot.temp500 <- get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500),1),count=c(length(xc_lm),length(yc_lm),1,1))
   pot.temp500[pot.temp500 == Fillval_temp] <- NA
   temp500 <- pot.temp500*(500/1000)^0.286
   difftemp500 <- SST_mean - temp500

   pot.temp700 <- get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700),1),count=c(length(xc_lm),length(yc_lm),1,1))
   pot.temp700[pot.temp700 == Fillval_temp] <- NA
   temp700 <- pot.temp700*(700/1000)^0.286
   difftemp700 <- SST_mean - temp700

   #Pressure (Maybe only in points?)
   Sea.level.pressure <- get.var.ncdf(temp_hir_1.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),length(yc_lm),1))

   close.ncdf(temp_hir_1.nc)
  }
 else{
  difftemp500 <- matrix(nrow=2150,ncol=1650)
  difftemp700 <- matrix(nrow=2150,ncol=1650)
  Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
 }

 missval = 1.0e10
 difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
 difftemp500[is.na(difftemp500)] = missval
 difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
 difftemp700[is.na(difftemp700)] = missval


   # Save as netcdf file
   outfile = paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")
 cat(paste(outfile,"\n"))

 xc <- dim.def.ncdf("xc","km",1:2150)
 yc <- dim.def.ncdf("yc","km",1:1650)
   varAHP500 <- var.def.ncdf("ahp_500", "degrees", list(xc,yc), missval, longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
   varAHP700 <- var.def.ncdf("ahp_700", "degrees", list(xc,yc), missval, longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
   varPres <- var.def.ncdf("sea_level_pressure", "Pa",list(xc,yc),-1, longname="Sea level pressure", prec="float")

   # Create a netCDF file with this variable
 ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )
```

```
    # Write some values to this variable on disk.
  put.var.ncdf( ncnew, varAHP500, difftemp500)
  put.var.ncdf( ncnew, varAHP700, difftemp700)
  put.var.ncdf( ncnew, varPres, Sea.level.pressure)

  close.ncdf(ncnew)


}


#The Autumn
07!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
start.time.A07 <- ISOdatetime(2007,10,1,0,0,0,tz="UTC")
startjulday.A07 <- strptime(start.time.A07,"%Y-%m-%d")$yday+1
endjulday.A07 <- strptime(ISOdatetime(2007,12,31,0,0,0,tz="UTC"),"%Y-%m-%d")$yday+1

for( i in startjulday.A07:endjulday.A07) {
  cat(paste("processing julian day ",i," in 2007\n",sep=""))

  epocl <- start.time.A07 + 24*60*60*(i-startjulday.A07)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")

  #SST----------------------------------------------------------------------------------------------------------------------------------------------
  filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
   if(length(filelist_sst_1) != 0) {
    for(j in 1:length(filelist_sst_1)) {

      temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

      xc <- get.var.ncdf(temp_sst_1.nc,"xc")
      yc <- get.var.ncdf(temp_sst_1.nc,"yc")
      sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                                #get fillvalue for SST
         Fillval_sst = att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

    #set sst with fillvalue to NA's
      sst[sst==Fillval_sst] <- NA

                                #Total vector with SST:
      if( j == 1) {
        SST_tot <- as.vector(sst)
      }
      else{
        SST_tot <- rbind(SST_tot,as.vector(sst))
      }
      close.ncdf(temp_sst_1.nc)
    }
```

```
                              #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc
   if(length(filelist_sst_1) != 1) {
    SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
   }
   else{
    SST_mean <- matrix(SST_tot,dim(sst))
   }

  }
  else {
   SST_mean <- matrix(nrow=2150,ncol=1650)
   cat(paste("No SST-files for julian day ",i," 2007","\n",sep=""))
  }

                                             #T500                    /                    T700
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------

  file.hir.1 <- paste(file.loc1,"/","h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

  if( file.exists(file.hir.1) ){
    temp_hir_1.nc <- open.ncdf(file.hir.1)
    timevector <- get.var.ncdf(temp_hir_1.nc,"time")
    pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")

   #Temp_diff
    Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value

                                                                    pot.temp500              <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp500[pot.temp500 == Fillval_temp] <- NA
    temp500 <- pot.temp500*(500/1000)^0.286
    difftemp500 <- SST_mean - temp500

                                                                    pot.temp700              <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp700[pot.temp700 == Fillval_temp] <- NA
    temp700 <- pot.temp700*(700/1000)^0.286
    difftemp700 <- SST_mean - temp700

    #Pressure (Maybe only in points?)
                                                              Sea.level.pressure             <-
get.var.ncdf(temp_hir_1.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),length
(yc_lm),1))

    close.ncdf(temp_hir_1.nc)
   }
  else{
   difftemp500 <- matrix(nrow=2150,ncol=1650)
   difftemp700 <- matrix(nrow=2150,ncol=1650)
   Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
  }
```

```
  missval = 1.0e10
  difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
  difftemp500[is.na(difftemp500)] = missval
  difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
  difftemp700[is.na(difftemp700)] = missval

    # Save as netcdf file
                                                            outfile                                    =
paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h20pl00_",format(epocl,"%Y
%m%d"),".nc",sep="")
  cat(paste(outfile,"\n"))

  xc <- dim.def.ncdf("xc","km",1:2150)
  yc <- dim.def.ncdf("yc","km",1:1650)
        varAHP500    <-    var.def.ncdf("ahp_500",    "degrees",    list(xc,yc),    missval,
longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
        varAHP700    <-    var.def.ncdf("ahp_700",    "degrees",    list(xc,yc),    missval,
longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
   varPres  <-  var.def.ncdf("sea_level_pressure",  "Pa",  list(xc,yc),  -1,  longname="Sea  level
pressure", prec="float")

    # Create a netCDF file with this variable
  ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )

     # Write some values to this variable on disk.
  put.var.ncdf( ncnew, varAHP500, difftemp500)
  put.var.ncdf( ncnew, varAHP700, difftemp700)
  put.var.ncdf( ncnew, varPres, Sea.level.pressure)

  close.ncdf(ncnew)


}

#The       Spring       08       until       day       43       when       h12
begins!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
start.time.08 <- ISOdatetime(2008,1,1,0,0,0,tz="UTC")
startjulday.08 <- strptime(start.time.08,"%Y-%m-%d")$yday+1
endjulday.08 <- strptime(ISOdatetime(2008,04,30,0,0,0,tz="UTC"),"%Y-%m-%d")$yday+1

for( i in startjulday.08:43) {
  cat(paste("processing julian day ",i," in 2008\n",sep=""))

  epocl <- start.time.08 + 24*60*60*(i-startjulday.08)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")

   #SST----------------------------------------------------------------------------------------------------------
-------------------------------------
  filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
   if(length(filelist_sst_1) != 0) {
    for(j in 1:length(filelist_sst_1)) {
```

```
    temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

    xc <- get.var.ncdf(temp_sst_1.nc,"xc")
    yc <- get.var.ncdf(temp_sst_1.nc,"yc")
    sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                              #get fillvalue for SST
        Fillval_sst     =  att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

   #set sst with fillvalue to NA's
    sst[sst==Fillval_sst] <- NA

                              #Total vector with SST:
    if( j == 1) {
      SST_tot <- as.vector(sst)
    }
    else{
      SST_tot <- rbind(SST_tot,as.vector(sst))
    }
    close.ncdf(temp_sst_1.nc)
   }

                              #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc
   if(length(filelist_sst_1) != 1) {
    SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
   }
   else{
    SST_mean <- matrix(SST_tot,dim(sst))
   }

  }
  else {
   SST_mean <- matrix(nrow=2150,ncol=1650)
   cat(paste("No SST-files for julian day ",i," 2008","\n",sep=""))
  }

                                             #T500                  /                  T700
--------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------

  file.hir.1 <- paste(file.loc1,"/","h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

  if( file.exists(file.hir.1) ){
    temp_hir_1.nc <- open.ncdf(file.hir.1)
    timevector <- get.var.ncdf(temp_hir_1.nc,"time")
    pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")

   #Temp_diff
    Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value

                                                            pot.temp500              <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
```

```
    pot.temp500[pot.temp500 == Fillval_temp] <- NA
    temp500 <- pot.temp500*(500/1000)^0.286
    difftemp500 <- SST_mean - temp500

    pot.temp700 <- get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700),1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp700[pot.temp700 == Fillval_temp] <- NA
    temp700 <- pot.temp700*(700/1000)^0.286
    difftemp700 <- SST_mean - temp700

    #Pressure (Maybe only in points?)
    Sea.level.pressure <- get.var.ncdf(temp_hir_1.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),length(yc_lm),1))

    close.ncdf(temp_hir_1.nc)
   }
  else{
   difftemp500 <- matrix(nrow=2150,ncol=1650)
   difftemp700 <- matrix(nrow=2150,ncol=1650)
   Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
  }

  missval = 1.0e10
  difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
  difftemp500[is.na(difftemp500)] = missval
  difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
  difftemp700[is.na(difftemp700)] = missval


    # Save as netcdf file
    outfile = paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h20pl00_",format(epocl,"%Y%m%d"),".nc",sep="")
  cat(paste(outfile,"\n"))

  xc <- dim.def.ncdf("xc","km",1:2150)
  yc <- dim.def.ncdf("yc","km",1:1650)
    varAHP500 <- var.def.ncdf("ahp_500", "degrees", list(xc,yc), missval, longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
    varAHP700 <- var.def.ncdf("ahp_700", "degrees", list(xc,yc), missval, longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
    varPres <- var.def.ncdf("sea_level_pressure", "Pa",list(xc,yc),-1, longname="Sea level pressure", prec="float")

    # Create a netCDF file with this variable
  ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )


     # Write some values to this variable on disk.
  put.var.ncdf( ncnew, varAHP500, difftemp500)
  put.var.ncdf( ncnew, varAHP700, difftemp700)
  put.var.ncdf( ncnew, varPres, Sea.level.pressure)
```

```
  close.ncdf(ncnew)


}

#Juldays in spring 08 after day
43:!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!

for( i in 44:endjulday.08) {
  cat(paste("processing julian day ",i," in 2008\n",sep=""))

  epocl <- start.time.08 + 24*60*60*(i-startjulday.08)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")

   #SST---------------------------------------------------------------------------------------------------------------
--------------------------------------
  filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
   if(length(filelist_sst_1) != 0) {
    for(j in 1:length(filelist_sst_1)) {

      temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

      xc <- get.var.ncdf(temp_sst_1.nc,"xc")
      yc <- get.var.ncdf(temp_sst_1.nc,"yc")
      sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                                  #get fillvalue for SST
           Fillval_sst      =  att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

    #set sst with fillvalue to NA's
      sst[sst==Fillval_sst] <- NA

                                  #Total vector with SST:
      if( j == 1) {
        SST_tot <- as.vector(sst)
      }
      else{
        SST_tot <- rbind(SST_tot,as.vector(sst))
      }
      close.ncdf(temp_sst_1.nc)
    }

                                  #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc
    if(length(filelist_sst_1) != 1) {
      SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
    }
    else{
      SST_mean <- matrix(SST_tot,dim(sst))
    }

  }
```

```
 else {
  SST_mean <- matrix(nrow=2150,ncol=1650)
  cat(paste("No SST-files for julian day ",i," 2008","\n",sep=""))
 }

                                                    #T500                    /                    T700
----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------

 file.hir.1 <- paste(file.loc1,"/","h12pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

 if( file.exists(file.hir.1) ){
   temp_hir_1.nc <- open.ncdf(file.hir.1)
   timevector <- get.var.ncdf(temp_hir_1.nc,"time")
   pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")

  #Temp_diff
   Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value

                                                                    pot.temp500                  <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
   pot.temp500[pot.temp500 == Fillval_temp] <- NA
   temp500 <- pot.temp500*(500/1000)^0.286
   difftemp500 <- SST_mean - temp500

                                                                    pot.temp700                  <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
   pot.temp700[pot.temp700 == Fillval_temp] <- NA
   temp700 <- pot.temp700*(700/1000)^0.286
   difftemp700 <- SST_mean - temp700

   close.ncdf(temp_hir_1.nc)
  }
 else{
  difftemp500 <- matrix(nrow=2150,ncol=1650)
  difftemp700 <- matrix(nrow=2150,ncol=1650)
 }

 file.hir.Press <- paste(file.loc1,"/","h12sf00_",format(epocl,"%Y%m%d"),".nc",sep="")

  if (file.exists(file.hir.Press) ) {
   temp.hir.Press.nc <- open.ncdf(file.hir.Press)
                                                              Sea.level.pressure                  <-
get.var.ncdf(temp.hir.Press.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),len
gth(yc_lm),1))
   close.ncdf( temp.hir.Press.nc)
  }
 else{
  Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
 }

 missval = 1.0e10
 difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
```

```
  difftemp500[is.na(difftemp500)] = missval
  difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
  difftemp700[is.na(difftemp700)] = missval


    # Save as netcdf file
                                                                                      outfile                         =
paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h12pl00_",format(epocl,"%Y
%m%d"),".nc",sep="")
  cat(paste(outfile,"\n"))

  xc <- dim.def.ncdf("xc","km",1:2150)
  yc <- dim.def.ncdf("yc","km",1:1650)
          varAHP500     <-     var.def.ncdf("ahp_500",     "degrees",     list(xc,yc),     missval,
longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
          varAHP700     <-     var.def.ncdf("ahp_700",     "degrees",     list(xc,yc),     missval,
longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
    varPres  <-  var.def.ncdf("sea_level_pressure",  "Pa",list(xc,yc),-1,  longname="Sea  level
pressure", prec="float")

    # Create a netCDF file with this variable
  ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )


     # Write some values to this variable on disk.
  put.var.ncdf( ncnew, varAHP500, difftemp500)
  put.var.ncdf( ncnew, varAHP700, difftemp700)
  put.var.ncdf( ncnew, varPres, Sea.level.pressure)

  close.ncdf(ncnew)


}


#Autumn
08!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

start.time.A08 <- ISOdatetime(2008,10,1,0,0,0,tz="UTC")
startjulday.A08 <- strptime(start.time.A08,"%Y-%m-%d")$yday+1
endjulday.A08 <- strptime(ISOdatetime(2008,12,31,0,0,0,tz="UTC"),"%Y-%m-%d")$yday+1

for( i in startjulday.A08:endjulday.A08) {
  cat(paste("processing julian day ",i," in 2008\n",sep=""))

  epocl <- start.time.A08 + 24*60*60*(i-startjulday.A08)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")

   #SST--------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
  filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
   if(length(filelist_sst_1) != 0) {
    for(j in 1:length(filelist_sst_1)) {
```

```
    temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

    xc <- get.var.ncdf(temp_sst_1.nc,"xc")
    yc <- get.var.ncdf(temp_sst_1.nc,"yc")
    sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                                    #get fillvalue for SST
         Fillval_sst      =  att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

   #set sst with fillvalue to NA's
    sst[sst==Fillval_sst] <- NA

                                    #Total vector with SST:
    if( j == 1) {
      SST_tot <- as.vector(sst)
    }
    else{
      SST_tot <- rbind(SST_tot,as.vector(sst))
    }
    close.ncdf(temp_sst_1.nc)
   }

                                    #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc
   if(length(filelist_sst_1) != 1) {
    SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
   }
   else{
    SST_mean <- matrix(SST_tot,dim(sst))
   }

  }
  else {
   SST_mean <- matrix(nrow=2150,ncol=1650)
   cat(paste("No SST-files for julian day ",i," 2008","\n",sep=""))
  }

                                                  #T500                    /                    T700
-------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------

  file.hir.1 <- paste(file.loc1,"/","h12pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

  if( file.exists(file.hir.1) ){
    temp_hir_1.nc <- open.ncdf(file.hir.1)
    timevector <- get.var.ncdf(temp_hir_1.nc,"time")
    pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")

   #Temp_diff
    Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value
```

```
pot.temp500 <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp500[pot.temp500 == Fillval_temp] <- NA
    temp500 <- pot.temp500*(500/1000)^0.286
    difftemp500 <- SST_mean - temp500

pot.temp700 <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
    pot.temp700[pot.temp700 == Fillval_temp] <- NA
    temp700 <- pot.temp700*(700/1000)^0.286
    difftemp700 <- SST_mean - temp700

    close.ncdf(temp_hir_1.nc)
  }
  else{
   difftemp500 <- matrix(nrow=2150,ncol=1650)
   difftemp700 <- matrix(nrow=2150,ncol=1650)
  }

  file.hir.Press <- paste(file.loc1,"/","h12sf00_",format(epocl,"%Y%m%d"),".nc",sep="")

   if (file.exists(file.hir.Press) ) {
    temp.hir.Press.nc <- open.ncdf(file.hir.Press)
Sea.level.pressure <-
get.var.ncdf(temp.hir.Press.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),len
gth(yc_lm),1))
    close.ncdf( temp.hir.Press.nc)
   }
  else{
   Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
  }

  missval = 1.0e10
  difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
  difftemp500[is.na(difftemp500)] = missval
  difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
  difftemp700[is.na(difftemp700)] = missval


    # Save as netcdf file
outfile =
paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h12pl00_",format(epocl,"%Y
%m%d"),".nc",sep="")
  cat(paste(outfile,"\n"))

  xc <- dim.def.ncdf("xc","km",1:2150)
  yc <- dim.def.ncdf("yc","km",1:1650)
        varAHP500 <- var.def.ncdf("ahp_500", "degrees", list(xc,yc), missval,
longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
        varAHP700 <- var.def.ncdf("ahp_700", "degrees", list(xc,yc), missval,
longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
    varPres <- var.def.ncdf("sea_level_pressure", "Pa",list(xc,yc),-1, longname="Sea level
pressure", prec="float")
```

```
    # Create a netCDF file with this variable
  ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )

    # Write some values to this variable on disk.
  put.var.ncdf( ncnew, varAHP500, difftemp500)
  put.var.ncdf( ncnew, varAHP700, difftemp700)
  put.var.ncdf( ncnew, varPres, Sea.level.pressure)

  close.ncdf(ncnew)

}


#Spring
09!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


start.time.09 <- ISOdatetime(2009,1,1,0,0,0,tz="UTC")
startjulday.09 <- strptime(start.time.09,"%Y-%m-%d")$yday+1
endjulday.09 <- strptime(ISOdatetime(2009,04,30,0,0,0,tz="UTC"),"%Y-%m-%d")$yday+1

for( i in startjulday.09:endjulday.09) {
  cat(paste("processing julian day ",i," in 2009\n",sep=""))

  epocl <- start.time.09 + 24*60*60*(i-startjulday.09)
  thisjulday <- formatC(strptime(epocl,"%Y-%m-%d")$yday+1,width=3,flag="0")
  file.loc1 <- paste(startstring,format(epocl,"%Y"),thisjulday,sep="/")

   #SST-------------------------------------------------------------------------------------------------------------------
--------------------------------------
  filelist_sst_1 <- list.files(file.loc1,pattern="AMSRE-REMSS")
   if(length(filelist_sst_1) != 0) {
    for(j in 1:length(filelist_sst_1)) {

      temp_sst_1.nc <- open.ncdf(paste(file.loc1,filelist_sst_1[j],sep="/"))

      xc <- get.var.ncdf(temp_sst_1.nc,"xc")
      yc <- get.var.ncdf(temp_sst_1.nc,"yc")
      sst <- get.var.ncdf(temp_sst_1.nc,"sea_surface_temperature")

                              #get fillvalue for SST
            Fillval_sst      =  att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","_FillValue")
$value*att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","scale_factor")
$value+att.get.ncdf(temp_sst_1.nc,"sea_surface_temperature","add_offset")$value

    #set sst with fillvalue to NA's
      sst[sst==Fillval_sst] <- NA

                              #Total vector with SST:
      if( j == 1) {
        SST_tot <- as.vector(sst)
      }
      else{
```

```
      SST_tot <- rbind(SST_tot,as.vector(sst))
     }
     close.ncdf(temp_sst_1.nc)
    }

                                 #Daily mean SST-matrix in each gridpoint:  dimension = xc times yc
    if(length(filelist_sst_1) != 1) {
     SST_mean <- matrix(colMeans(SST_tot,na.rm=TRUE),dim(sst))
    }
    else{
     SST_mean <- matrix(SST_tot,dim(sst))
    }

  }
  else {
    SST_mean <- matrix(nrow=2150,ncol=1650)
    cat(paste("No SST-files for julian day ",i," 2009","\n",sep=""))
  }

                                                 #T500                      /                      T700
-------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------

  file.hir.1 <- paste(file.loc1,"/","h12pl00_",format(epocl,"%Y%m%d"),".nc",sep="")

  if( file.exists(file.hir.1) ){
     temp_hir_1.nc <- open.ncdf(file.hir.1)
     timevector <- get.var.ncdf(temp_hir_1.nc,"time")
     pressurevec <- get.var.ncdf(temp_hir_1.nc,"pressure")

    #Temp_diff
     Fillval_temp <- att.get.ncdf(temp_hir_1.nc,"air_potential_temperature","_FillValue")$value

                                                                       pot.temp500                  <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==500)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
     pot.temp500[pot.temp500 == Fillval_temp] <- NA
     temp500 <- pot.temp500*(500/1000)^0.286
     difftemp500 <- SST_mean - temp500

                                                                       pot.temp700                  <-
get.var.ncdf(temp_hir_1.nc,"air_potential_temperature",start=c(1,1,which(pressurevec==700)
,1),count=c(length(xc_lm),length(yc_lm),1,1))
     pot.temp700[pot.temp700 == Fillval_temp] <- NA
     temp700 <- pot.temp700*(700/1000)^0.286
     difftemp700 <- SST_mean - temp700

     close.ncdf(temp_hir_1.nc)
   }
  else{
    difftemp500 <- matrix(nrow=2150,ncol=1650)
    difftemp700 <- matrix(nrow=2150,ncol=1650)
  }

  file.hir.Press <- paste(file.loc1,"/","h12sf00_",format(epocl,"%Y%m%d"),".nc",sep="")
```

```
  if (file.exists(file.hir.Press) ) {
    temp.hir.Press.nc <- open.ncdf(file.hir.Press)
    Sea.level.pressure <-
get.var.ncdf(temp.hir.Press.nc,"sea_level_pressure",start=c(1,1,1),count=c(length(xc_lm),length(yc_lm),1))
    close.ncdf( temp.hir.Press.nc)
  }
 else{
   Sea.level.pressure <- matrix(nrow=2150,ncol=1650)
 }

 missval = 1.0e10
 difftemp500[difftemp500 < -999.9 | difftemp500 > 999.9] = missval
 difftemp500[is.na(difftemp500)] = missval
 difftemp700[difftemp700 < -999.9 | difftemp700 > 999.9] = missval
 difftemp700[is.na(difftemp700)] = missval


   # Save as netcdf file
   outfile =
paste(startstring,"/hirlam/",format(epocl,"%Y"),"/",thisjulday,"/pli_h12pl00_",format(epocl,"%Y%m%d"),".nc",sep="")
 cat(paste(outfile,"\n"))

 xc <- dim.def.ncdf("xc","km",1:2150)
 yc <- dim.def.ncdf("yc","km",1:1650)
     varAHP500 <- var.def.ncdf("ahp_500", "degrees", list(xc,yc), missval,
longname="Atmospheric Heat Potential SST-T_500hPa", prec="float")
     varAHP700 <- var.def.ncdf("ahp_700", "degrees", list(xc,yc), missval,
longname="Atmospheric Heat Potential SST-T_700hPa", prec="float")
   varPres <- var.def.ncdf("sea_level_pressure", "Pa",list(xc,yc),-1, longname="Sea level
pressure", prec="float")

   # Create a netCDF file with this variable
 ncnew <- create.ncdf( outfile, list(varAHP500,varAHP700,varPres) )

    # Write some values to this variable on disk.
 put.var.ncdf( ncnew, varAHP500, difftemp500)
 put.var.ncdf( ncnew, varAHP700, difftemp700)
 put.var.ncdf( ncnew, varPres, Sea.level.pressure)

 close.ncdf(ncnew)

}
```